**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ЛАБОРАТОРНАЯ РАБОТА №9**

по дисциплине: «Объектно-ориентированное программирование»

на тему: «Обработка исключительных ситуаций»

Выполнил: студент гр. ИТП-21

Леди Д.Д.

Принял: преподаватель-стажер

Гуменников Е.Д.

Гомель 2022

**Цель работы:** Изучить что такое абстрактные классы и интерфейсы. Изучить основные принципы наследования в *C*#. Реализовать оконное приложение при помощи библиотеки классов *WFA*.

**Задание**

1. Необходимо дополнить решение задачи из лабораторной работы №2

(таблица 2) обработкой всевозможных исключительных ситуаций и

обеспечить надёжное функционирование классов.

2. При обработке исключений использовать собственные классы

исключений, которые необходимо разместить в отдельной библиотеке

классов

3. При создании классов руководствоваться Code Convention

4. Весь код должен содержать элементы документирования

5. Разработать модульные тесты для верификации созданных классов

6. Классы должен быть размещены в библиотеке классов

7. Модульные тесты – в отдельном проекте

8. Дополнить задачу графическим интерфейсом: в отдельном проекте

реализовать для нечётных вариантов – WPF, для чётных – WFA.

**Ход Работы**

С начала был создан класс *Matrice* в котором были описаны основные методы и операторы перегрузки.

После этого был создан меню под управлением *WindowsForm.* Пример работы форм указаны на рисунке 1 – 4.

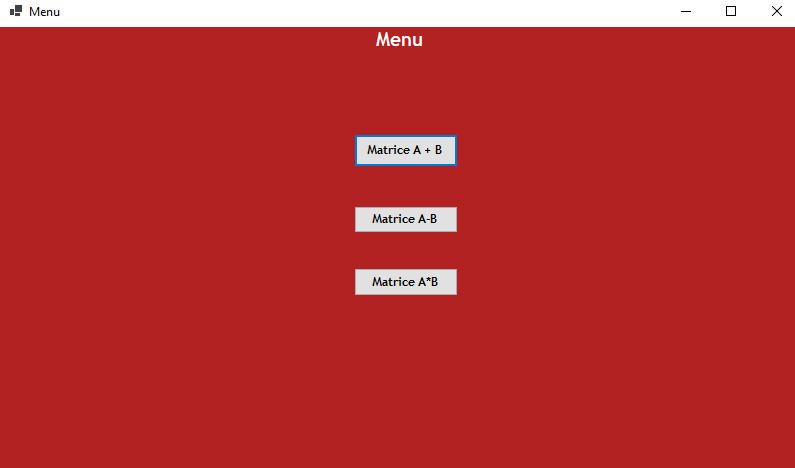


Рисунок 1 – Пример Form1

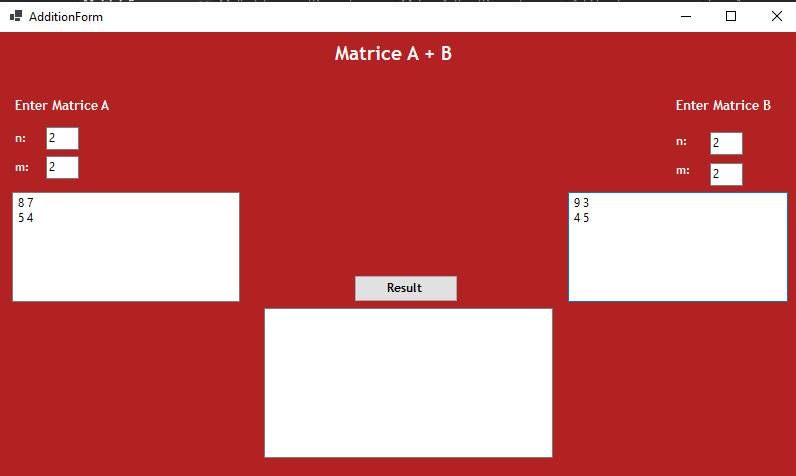


Рисунок 2 – Пример Form2

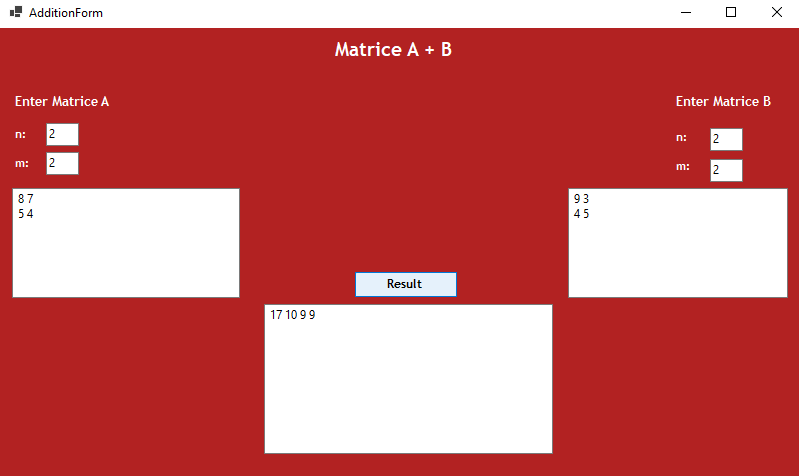


Рисунок 3 – Пример Form3

**Вывод:** в ходе выполнения лабораторной работы была изучена работа с *WFA.* Изучена такая концепция как наследование в языке программирования *C*#. И такие типы данных как абстрактные классы и интерфейсы.

**Листинг класса Matrice**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab9WinFormApp

{

public class Matrice

{

//field

public int[,] array { get; set; }

public int this[int i, int j]//l'indexeurs est utilisé pour accéder à une instance d’une classe

{

get

{

return array[i, j];

}

set

{

array[i, j] = value;

}

}

public int Lines

{

get

{

return array.GetLength(0);

}

}

public int Columns

{

get

{

return array.GetLength(1);

}

}

//constructor

public Matrice(int Lines, int Columns)

{

array = new int[Lines, Columns];

}

public static Matrice operator +(Matrice a, Matrice b)

{

Matrice c = new Matrice(a.Lines, a.Columns);

for (int i = 0; i < a.Lines; i++)

{

for (int j = 0; j < a.Columns; j++)

{

c[i, j] = a[i, j] + b[i, j];

}

}

return c;

}

public static Matrice operator -(Matrice a, Matrice b)

{

Matrice s = new Matrice(a.Lines, a.Columns);

for (int i = 0; i < a.Lines; i++)

{

for (int j = 0; j < a.Columns; j++)

{

s[i, j] = a[i, j] - b[i, j];

}

}

return s;

}

public static Matrice operator \*(Matrice a, Matrice b)

{

Matrice m = new Matrice(a.Lines, a.Columns);

for (int i = 0; i < a.Lines; i++)

{

for (int j = 0; j < a.Columns; j++)

{

m[i, j] = a[i, j] \* b[i, j];

}

}

return m;

}

}

}

**Листинг класса Form1**

namespace Lab9WinFormApp

{

public partial class Menu : Form

{

public Menu()

{

InitializeComponent();

}

private void AdditionBttn\_Click(object sender, EventArgs e)

{

var window = new AdditionForm();

window.Show();

}

private void SoustractBttn\_Click(object sender, EventArgs e)

{

var window = new MatriceA\_B();

window.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

var window = new MultiplyForm();

window.Show();

}

}

**Листинг класса AdditionForm**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Linq;

namespace Lab9WinFormApp

{

public partial class AdditionForm : Form

{

public AdditionForm()

{

InitializeComponent();

}

private void ResultBttnAdd\_Click(object sender, EventArgs e)

{

Try

{

int n1 = Convert.ToInt32(numberofLine.Text);

int m1 = Convert.ToInt32(numberofColumn.Text);

Matrice matriceA = FillArray(n1, m1, textBoxMatriceA.Text);

int n2 = Convert.ToInt32(numberofLineB.Text);

int m2 = Convert.ToInt32(numberofColumnB.Text);

Matrice matriceB = FillArray(n2, m2, textBoxMatriceB.Text);

Matrice addition = matriceA + matriceB;

for (int i = 0; i < addition.Lines; i++)

{

for (int j = 0; j < addition.Columns; j++) {

textBoxResultAdd.Text += addition[i, j].ToString() + " ";

}

textBoxResultAdd.Text += "\n";

}

Catch(Exception ex)

{

MessageBox.Show(“Did you filled very well??Please Enter a number”)

}

}

private Matrice FillArray(int n, int m, String arrayInString)

{

Matrice matriceA = new Matrice(n, m);

string values1 = arrayInString;

char[] separators = new char[] {' ','\n' };

string[] tmp1 = values1.Split(separators);

int k =0;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

matriceA[i, j] = Convert.ToInt32(tmp1[k]);

k++;

}

}

return matriceA;

}

}

}

**Листинг класса SoustractForm**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab9WinFormApp

{

public partial class MatriceA\_B : Form

{

public MatriceA\_B()

{

InitializeComponent();

}

private void ResultBttnSoustr\_Click(object sender, EventArgs e)

{

Try

{

int n1 = Convert.ToInt32(numberofLineA2.Text);

int m1 = Convert.ToInt32(numberofColumnA2.Text);

Matrice matriceA = FillArray(n1, m1, textBoxMatriceA2.Text);

int n2 = Convert.ToInt32(numberoflineB2.Text);

int m2 = Convert.ToInt32(numberofcolumnB2.Text);

Matrice matriceB = FillArray(n2, m2, textBoxMatriceB2.Text);

Matrice soustraction = matriceA - matriceB;

for (int i = 0; i < soustraction.Lines; i++)

{

for (int j = 0; j < soustraction.Columns; j++)

{

textBoxResultSoust.Text += soustraction[i, j].ToString() + " ";

}

textBoxResultSoust.Text += "\n";

}

Catch(Exception ex)

{

MessageBox.Show(“Did you filled very well??Please Enter a number”)

}

}

}

private Matrice FillArray(int n, int m, String arrayInString)

{

Matrice matriceA = new Matrice(n, m);

string values1 = arrayInString;

char[] separators = new char[] { ' ', '\n' };

string[] tmp1 = values1.Split(separators);

int k = 0;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

matriceA[i, j] = Convert.ToInt32(tmp1[k]);

k++;

}

}

return matriceA;

}

}

}

**Листинг класса MultiplyForm**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab9WinFormApp

{

public partial class MultiplyForm : Form

{

public MultiplyForm()

{

InitializeComponent();

}

private void ResultBttnMult\_Click\_1(object sender, EventArgs e)

{

Try

{

int n1 = Convert.ToInt32(numberofLineA3.Text);

int m1 = Convert.ToInt32(numberofColumnA3.Text);

Matrice matriceA = FillArray(n1, m1, textBoxMatriceA3.Text);

int n2 = Convert.ToInt32(numberoflineB3.Text);

int m2 = Convert.ToInt32(numberofcolumnB3.Text);

Matrice matriceB = FillArray(n2, m2, textBoxMatriceB3.Text);

Matrice multiplication = matriceA \* matriceB;

for (int i = 0; i < multiplication.Lines; i++)

{

for (int j = 0; j < multiplication.Columns; j++)

{

textBoxResultMult.Text += multiplication[i, j].ToString() + " ";

}

textBoxResultMult.Text += "\n";

}

Catch(Exception ex)

{

MessageBox.Show(“Did you filled very well??Please Enter a number”)

}

}

}

private Matrice FillArray(int n, int m, String arrayInString)

{

Matrice matriceA = new Matrice(n, m);

string values1 = arrayInString;

char[] separators = new char[] { ' ', '\n' };

string[] tmp1 = values1.Split(separators);

int k = 0;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

matriceA[i, j] = Convert.ToInt32(tmp1[k]);

k++;

}

}

return matriceA;

}

}

}